

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

1           1. (Currently amended) A method for encrypting data in a computer in  
2     communication with a volatile memory and non-volatile storage device,  
3     comprising:  
4         initiating a paging operation to move ~~encrypting~~ pages in the volatile  
5     memory to move to a swap file in the non-volatile storage device, wherein the  
6     non-volatile storage device is as part of a virtual addressing system;  
7         generating codes to use to encrypt and decrypt the pages, wherein the  
8     codes are permanently lost if the computer performs a boot operation;  
9         encrypting the pages in the volatile memory;  
10        moving the encrypted pages from the volatile memory to the swap file; and  
11        upon receiving a subsequent request to transfer the encrypted pages from  
12     the swap file to the volatile memory,  
13                 decrypting the encrypted pages in the swap file to move  
14         back into the volatile memory; and  
15                 moving the decrypted pages in the swap file back into the  
16         volatile memory.

1           2 (Canceled).

1           3. (Currently amended) The method of ~~claim 2~~claim 1, wherein the codes  
2     comprise a public/private key pair generated using a public key cryptography

3 algorithm, wherein one key of the pair is used to encrypt the pages moved to the  
4 swap file and the other key of the pair is used to decrypt the page when moving  
5 the page from the swap file to the volatile memory.

1 4 (Canceled).

1 | 5. (Currently amended) The method of ~~claim 2~~claim 1, wherein the codes  
2 are loaded into a non-swappable region of the volatile memory that is not moved  
3 to the swap file.

1 6. (Original) A method for encrypting files in a computer file system in  
2 communication with a volatile memory and a non-volatile storage device, wherein  
3 files in the file system are associated with groups, comprising:  
4 providing, for each group, a group identifier, a list of user identifiers of  
5 users allowed to access files in the group, and a first encryption code;  
6 receiving a second encryption code for one user identifier;  
7 receiving an input/output (I/O) request from a requesting user identifier  
8 with respect to a target file, wherein one second encryption code has been  
9 received for the user identifier;  
10 determining the group associated with the target file and the first  
11 encryption code for the group;  
12 if the I/O request is a write operation, then using the determined first  
13 encryption code to encrypt the target file to write the target file to the non-volatile  
14 storage device; and  
15 if the I/O request is a read operation to read the target file from the non-  
16 volatile storage device, then performing:  
17 (i) determining whether the requesting user identifier is in the list  
18 of the determined group; and

19 (ii) if the requesting user identifier is in the list, then using the  
20 second encryption code for the user identifier to decrypt the target file.

1 7. (Original) The method of claim 6, further comprising:  
2 for each group, generating a public and private encryption key pair using a  
3 public key encryption algorithm, wherein the first encryption code for the group is  
4 one of the generated public key or private key and the second encryption code is  
5 the other one of the public or private key generated for the group.

1 8. (Original) The method of claim 7, further comprising receiving a  
2 plurality of keys from the user, wherein each received key is used to decrypt files  
3 associated with one group identifier.

1 9. (Original) The method of claim 7, further comprising:  
2 generating an index entry in a non-swappable region in the volatile  
3 memory; and  
4 adding to the index entry the user identifier of the user that provided they  
5 key, the group identifier associated with the received key, and the received key.

1 10. (Original) The method of claim 9, wherein the index entry for the user  
2 identifier and group identifier is only generated if the user identifier is included in  
3 the list associated with the group identifier, and wherein the user identifier cannot  
4 perform a read access for the target file if there is no index entry for the group  
5 identifier associated with the target file and the user identifier.

1 11. (Original) The method of claim 9, wherein files read and decrypted  
2 from the non-volatile storage device are cached in the volatile memory, and  
3 wherein if the requested file is unencrypted in the cache, returning the

4 unencrypted file from the cache to the requesting user identifier if the requesting  
5 user identifier is in the list associated with the group identifier and there is one  
6 index entry for the user identifier and group identifier in the volatile memory.

1 | 12. (Currently amended) The method of ~~claim 1~~claim 6, wherein the  
2 second encryption code is accessed from a removable storage medium.

1 13. (Original) A method for encrypting files in a computer in  
2 communication with a volatile memory and non-volatile storage device,  
3 comprising;  
4 generating an encryption code to encrypt a file and a decryption code to  
5 decrypt one file encrypted with the encryption code;  
6 loading the decryption code into the volatile memory, wherein the  
7 decryption code is erased from the volatile memory when the computer reboots;  
8 encrypting files with the encryption code to transfer from the volatile  
9 memory to the non-volatile storage device; and  
10 decrypting files with the decryption code maintained in the volatile  
11 memory to transfer from the non-volatile storage device to the volatile memory.

1 14. (Original) The method of claim 13, further comprising:  
2 generating a new encryption and decryption codes when the computer  
3 reboots, wherein the new encryption code is used to transfer files from the volatile  
4 memory to the non-volatile storage device and wherein the new decryption code is  
5 used to transfer files from the non-volatile storage device to the volatile memory  
6 as part of a read operation.

1 15. (Original) The method of claim 13, wherein the decryption code is  
2 loaded into a non-swappable region of the volatile memory.

1           16. (Original) The method of claim 13, wherein the files are transferred  
2 between the volatile memory and non-volatile storage as part of a virtual memory  
3 paging operation.

1           17. (Currently amended) A system for encrypting data, comprising:  
2 a volatile memory;  
3 a non-volatile storage device, wherein data is capable of being transferred  
4 between the volatile memory and non-volatile storage device;  
5       means for initiating a paging operation to move encrypting pages in the  
6 volatile memory to move to a swap file in the non-volatile storage device, wherein  
7 the non-volatile storage device is as-part of a virtual addressing system;  
8       means for generating codes to use to encrypt and decrypt the pages,  
9 wherein the codes are permanently lost if the computer performs a boot operation;  
10       means for encrypting the pages in the volatile memory;  
11       means for moving the encrypted pages from the volatile memory to the  
12 swap file;  
13       means for receiving a subsequent request to transfer the encryhpted pages  
14 from the swap file to the volatile memory;  
15       means for decrypting the encrypted pages in the swap file ~~to move back~~  
16 ~~into the volatile memory;~~ and  
17       means for moving the decrypted pages ~~in the swap file back into the~~  
18 volatile memory.

1           18 (Canceled).

1           19. (Currently amended) The system of ~~claim 18~~claim 17, wherein the  
2 codes comprise a public/private key pair generated using a public key  
3 cryptography algorithm, wherein one key of the pair is used to encrypt the pages

4 moved to the swap file and the other key of the pair is used to decrypt the page  
5 when moving the page from the swap file to the volatile memory.

1 20 (Canceled).

1 | 21. (Currently amended) The system of ~~claim 18~~claim 17, further  
2 comprising:

3 means for loading the codes into a non-swappable region of the volatile  
4 memory that is not moved to the swap file.

1 22. A system for encrypting files, comprising:

2 a non-volatile storage device, wherein the non-volatile storage device  
3 includes a computer file system, wherein files in the file system are associated  
4 with groups.

5 means for providing, for each group, a group identifier, a list of user  
6 identifiers of users allowed to access files in the group, and a first encryption  
7 code;

8 means for receiving a second encryption code for one user identifier;

9 means for receiving an input/output (I/O) request from a requesting user  
10 identifier with respect to a target file, wherein one second encryption code has  
11 been received for the user identifier;

12 means for determining the group associated with the target file and the  
13 first encryption code for the group;

14 means for using the determined first encryption code to encrypt the target  
15 file to write the target file to the non-volatile storage device if the I/O request is a  
16 write operation; and

17 means for performing if the I/O request is a read operation to read the  
18 target file from the non-volatile storage device:

19 (i) determining whether the requesting user identifier is in the list  
20 for the determined group; and  
21 (ii) if the requesting user identifier is in the list, then using the  
22 second encryption code for the user identifier to decrypt the target file.

1 23. (Original) The system of claim 22, further comprising:  
2 means for generating, for each group, a public and private encryption key  
3 pair using a public key encryption algorithm, wherein the first encryption code for  
4 the group is one of the generated public key or private key and the second  
5 encryption code is the other one of the public or private key generated for the  
6 group.

1 24. (Original) The system of claim 23, further comprising:  
2 means for receiving a plurality of keys from the user, wherein each  
3 received key is used to decrypt files associated with one group identifier.

1 25. (Original) The system of claim 23, further comprising:  
2 means for generating an index entry in a non-swappable region in the  
3 volatile memory; and  
4 means for adding to the index entry the user identifier of the user that  
5 provided the key, the group identifier associated with the received key, and the  
6 received key.

1 26. (Original) The system of claim 25, wherein the index entry for the user  
2 identifier and group identifier is only generated if the user identifier is included in  
3 the list associated with the group identifier, and wherein the user identifier cannot  
4 perform a read access for the target file if there is no index entry for the group  
5 identifier associated with the target file and the user identifier.

1           27. (Original) The system of claim 25, wherein files read and decrypted  
2 from the non-volatile storage device are cached in the volatile memory, further  
3 comprising:

4           returning the unencrypted file from the cache to the requesting user  
5 identifier if the requested file is unencrypted in the cache and if the requesting  
6 user identifier is in the list associated with the group identifier and if there is one  
7 index entry for the user identifier and group identifier in the volatile memory.

1           28. (Original) The system of claim 22, wherein the second encryption code  
2 is accessed from a removable storage medium.

1           29. (Original) A system for encrypting files, comprising:

2           a volatile memory;

3           a non-volatile storage device, wherein data is capable of being transferred  
4 between the volatile memory and non-volatile storage device;

5           means for generating an encryption code to encrypt a file and a decryption  
6 code to decrypt one file encrypted with the encryption code;

7           means for loading the decryption code into the volatile memory, wherein  
8 the decryption code is erased from the volatile memory when the computer  
9 reboots;

10          means for encrypting files with the encryption code to transfer from the  
11 volatile memory to the non-volatile storage device; and

12          means for decrypting files with the decryption code maintained in the  
13 volatile memory to transfer from the non-volatile storage device to the volatile  
14 memory.

1           30. (Original) The system of claim 29, further comprising:



2 means for generating a new encryption and decryption codes when the  
3 computer reboots, wherein the new encryption code is used to transfer files from  
4 the volatile memory to the non-volatile storage device and wherein the new  
5 decryption code is used to transfer files from the non-volatile storage device to the  
6 volatile memory as part of a read operation.

1 31. (Original) The system of claim 29, wherein the decryption code is  
2 loaded into a non-swappable region of the volatile memory.

1 32. (Original) The system of claim 29, wherein the files are transferred  
2 between the volatile memory and non-volatile storage as part of a virtual memory  
3 paging operation.

1 33. (Currently amended) An article of manufacture including program  
2 logic for encrypting data in a computer in communication with a volatile memory  
3 and non-volatile storage device, by:

4 initiating a paging operation to move ~~encrypting~~ pages in the volatile  
5 memory to move to a swap file in the non-volatile storage device, wherein the  
6 non-volatile storage devise is as part of a virtual addressing system;  
7 generating codes to use to encrypt and decrypt the pages, wherein the  
8 codes are permanently lost if the computer performs a boot operation;  
9 encrypting the pages in the volatile memory;  
10 moving the encrypted pages from the volatile memory to the swap file; and  
11 upon receiving a subsequent request to transfer the encryhpted pages from  
12 the swap file to the volatile memory,  
13 decrypting the encrypted pages in the swap file to move  
14 back into the volatile memory; and

15 | moving the decrypted pages in the swap file back into the  
16 | volatile memory.

1 | 34 (Canceled).

1 | 35. (Currently amended) The article of manufacture of ~~claim 34~~claim 33,  
2 | wherein the codes comprise a public/private key pair generated using a public key  
3 | cryptography algorithm, wherein one key of the pair is used to encrypt the pages  
4 | moved to the swap file and the other key of the pair is used to decrypt the page  
5 | when moving the page from the swap file to the volatile memory.

1 | 36 (Canceled).

1 | 37. (Currently amended) The article of manufacture of ~~claim 34~~claim 33,  
2 | wherein the codes are loaded into a non-swappable region of the volatile memory  
3 | that is not moved to the swap file.

1 | 38. (Original) An article of manufacture including program logic for  
2 | encrypting files in a computer file system in communication with a volatile  
3 | memory and a non-volatile storage device, wherein files in the file system are  
4 | associated with groups by:  
5 | providing, for each group, a group identifier, a list of user identifiers of  
6 | users allowed to access files in the group, and a first encryption code;  
7 | receiving a second encryption code for one user identifier;  
8 | receiving an input/output (I/O) request from a requesting user identifier  
9 | with respect to a target file, wherein one second encryption code has been  
10 | received for the user identifier;

11           determining the group associated with the target file and the first  
12 encryption code for the group;  
13           if the I/O request is a write operation, then using the determined first  
14 encryption code to encrypt the target file to write the target file to the non-volatile  
15 storage device; and  
16           if the I/O request is a read operation to read the target file from the non-  
17 volatile storage device, then performing:  
18               (i) determining whether the requesting user identifier is in the list  
19               for the determined group; and  
20               (ii) if the requesting user identifier is in the list, then using the  
21 second encryption code for the user identifier to decrypt the target file.

1           39. (Original) The article of manufacture of claim 38, further comprising:  
2           for each group, generating a public and private encryption key pair using a  
3 public key encryption algorithm, wherein the first encryption code for the group is  
4 one of the generated public key or private key and the second encryption code is  
5 the other one of the public or private key generated for the group.

1           40. (Original) The article of manufacture of claim 39, further comprising  
2 receiving a plurality of keys from the user, wherein each received key is used to  
3 decrypt files associated with one group identifier.

1           41. (Original) The article of manufacture of claim 39, further comprising:  
2           generating an index entry in a non-swappable region in the volatile  
3 memory; and  
4           adding to the index entry the user identifier of the user that provided the  
5 key, the group identifier associated with the received key, and the received key.

1           42. (Original) The article of manufacture of claim 41, wherein the index  
2 entry for the user identifier and group identifier is only generated if the user  
3 identifier is included in the list associated with the group identifier, and wherein  
4 the user identifier cannot perform a read access for the target file if there is no  
5 index entry for the group identifier associated with the target file and the user  
6 identifier.

1           43. (Original) The article of manufacture of claim 41, wherein files read  
2 and decrypted from the non-volatile storage device are cached in the volatile  
3 memory, and wherein if the requested file is unencrypted in the cache, returning  
4 the unencrypted file from the cache to the requesting user identifier if the  
5 requesting user identifier is in the list associated with the group identifier and  
6 there is one index entry for the user identifier and group identifier in the volatile  
7 memory.

1           44. (Original) The article of manufacture of claim 38, wherein the second  
2 encryption code is accessed from a removable storage medium.

1           45. (Original) An article of manufacture including program logic for  
2 encrypting files in a computer in communication with a volatile memory and non-  
3 volatile storage device by:

4           generating an encryption code to encrypt a file and a decryption code to  
5 decrypt one file encrypted with the encryption code;

6           loading the decryption code into the volatile memory, wherein the  
7 decryption code is erased from the volatile memory when the computer reboots;

8           encrypting files with the encryption code to transfer from the volatile  
9 memory to the non-volatile storage device; and

10            decrypting files with the decryption code maintained in the volatile  
11 memory to transfer from the non-volatile storage device to the volatile memory.

1            46. (Original) The article of manufacture of claim 45, further comprising:  
2            generating a new encryption and decryption codes when the computer  
3 reboots, wherein the new encryption code is used to transfer files from the volatile  
4 memory to the non-volatile storage device and wherein the new decryption code is  
5 used to transfer files from the non-volatile storage device to the volatile memory  
6 as part of a read operation.

1            47. (Original) The article of manufacture of claim 45, wherein the  
2 decryption code is loaded into a non-swappable region of the volatile memory.

1            48. (Original) The article of manufacture of claim 45, wherein the files are  
2 transferred between the volatile memory and non-volatile storage as part of a  
3 virtual memory paging operation.